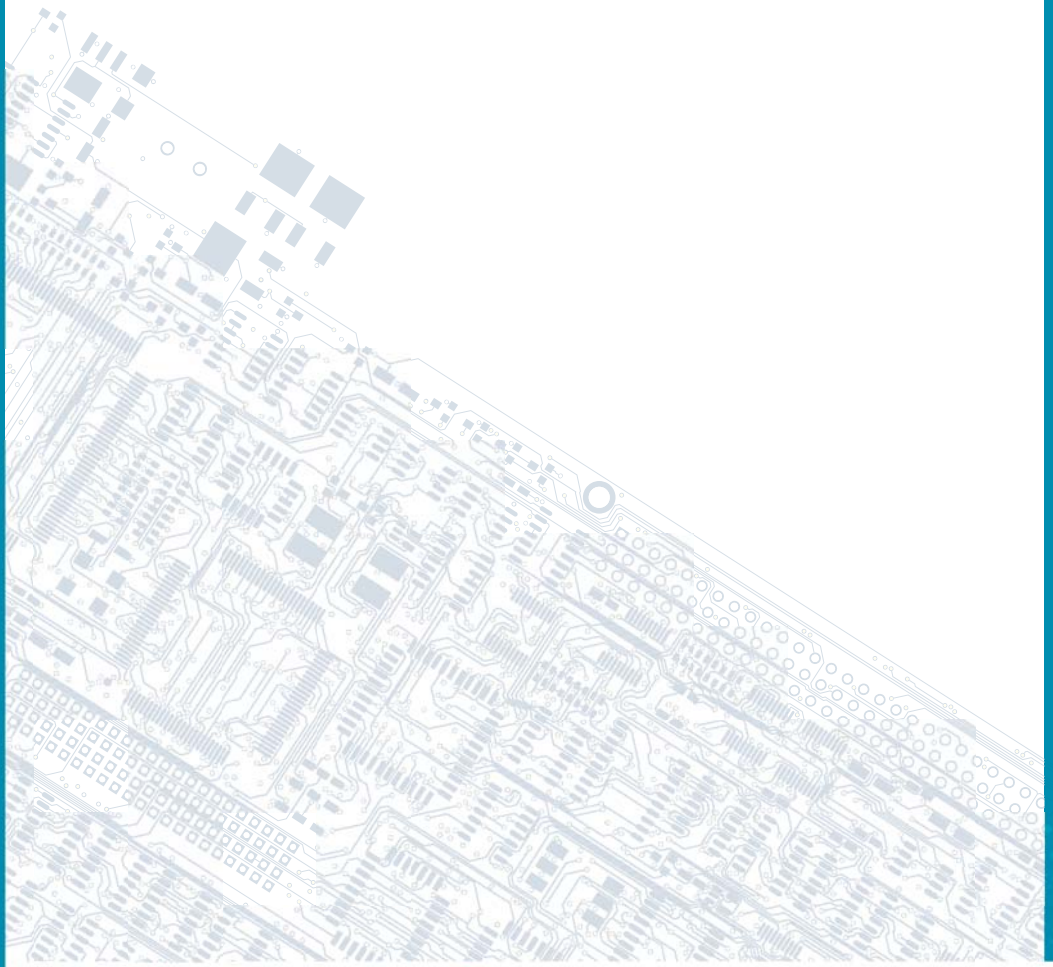
Easy 8051 v6™

User manual

All Mikroelektronika´s development systems represent irreplaceable tools for programming and developing microcontroller-based devices. Carefully chosen components and the use of machines of the last generation for mounting and testing thereof are the best guarantee of high reliability of our devices. Due to simple design, a large number of add-on modules and ready to use examples, all our users, regardless of their experience, have the possibility to develop their project in a fast and efficient way.

Development System



MikroElektronika

SOFTWARE AND HARDWARE SOLUTIONS FOR EMBEDDED WORLD ...making it simple

TO OUR VALUED CUSTOMERS

I want to express my thanks to you for being interested in our products and for having confidence in mikroElektronika.
The primary aim of our company is to design and produce high quality electronic products and to constantly improve the performance thereof in order to better suit your needs.

Nebojsa Matic
General Manager

The Atmel name and logo, the Atmel logo, AVR, AVR (Logo), AVR Freaks, AVR Freaks (Logo), AVR Studio, IDIC, megaAVR, megaAVR (Logo), picoPower ®, tinyAVR ® are trademarks of Atmel Coorporation.

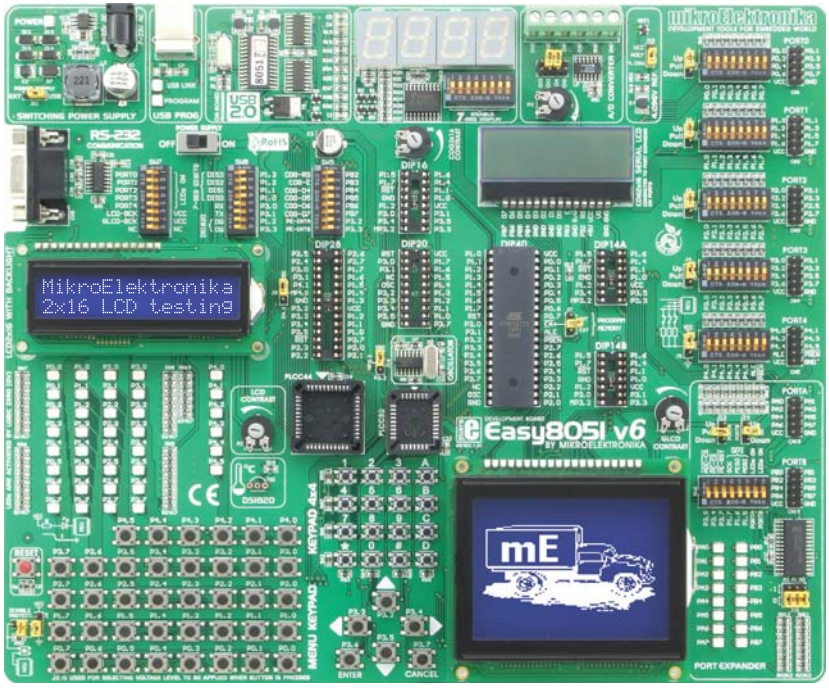
TABLE OF CONTENTS

Introduction to Easy8051 v6 Development System 4
Key Features 5
1.0. Connecting the System to your PC 6
2.0. Supported Microcontrollers 7
3.0. On-board USB 2.0 8051prog Programmer 8
4.0. Clock Oscillator 9
5.0 Power Supply 10
6.0 RS-232 Communication Interface 11
7.0. A/D Converter 12
8.0. DS1820 Temperature Sensor 13
9.0. LEDs 14
10.0. Push Buttons 15
11.0. MENU Keypad 16
12.0. 4x4 Keypad 17
13.0. On-board 2x16 LCD Display 18
14.0. 2x16 LCD Display 19
15.0. 128x64 Graphic LCD Display 20
16.0. Seven-Segment Display 21
17.0. I/O Ports 23
18.0. Port Expander (Additional I/O Ports) 25

page

Introduction to Easy8051 v6 Development System

The *Easy8051 v6*™ development system is a fully contained board suitable for programming 8051 microcontrollers from Atmel as well as for designing and testing 8051 projects. This development system includes an on-board programmer providing an interface between the microcontroller and a PC. You are simply expected to write a program in one of the 8051 compilers, generate a .hex file and program your microcontroller using the on-board *8051prog*™ programmer. The *Easy8051 v6* contains many devices, such as 128x64 graphic LCD display, alphanumeric 2x16 LCD display, on-board 2x16 LCD display, 4x4 keypad, port expander etc, that conveniently interface with 8051 microcontrollers and allow you to easily simulate the operation of the target device.



8051 DEVELOPMENT BOARD — Full-featured and user-friendly development system for 8051 microcontroller based devices

USB 2.0 IN-CIRCUIT PROGRAMMER — High-performance on-board USB 2.0 programmer

PORT EXPANDER — Port expander gives two more I/O ports to your 8051 microcontroller

COG 2x16 SERIAL LCD — Serial on-board 2x16 LCD display

GLCD128x64 — Graphic LCD display with backlight

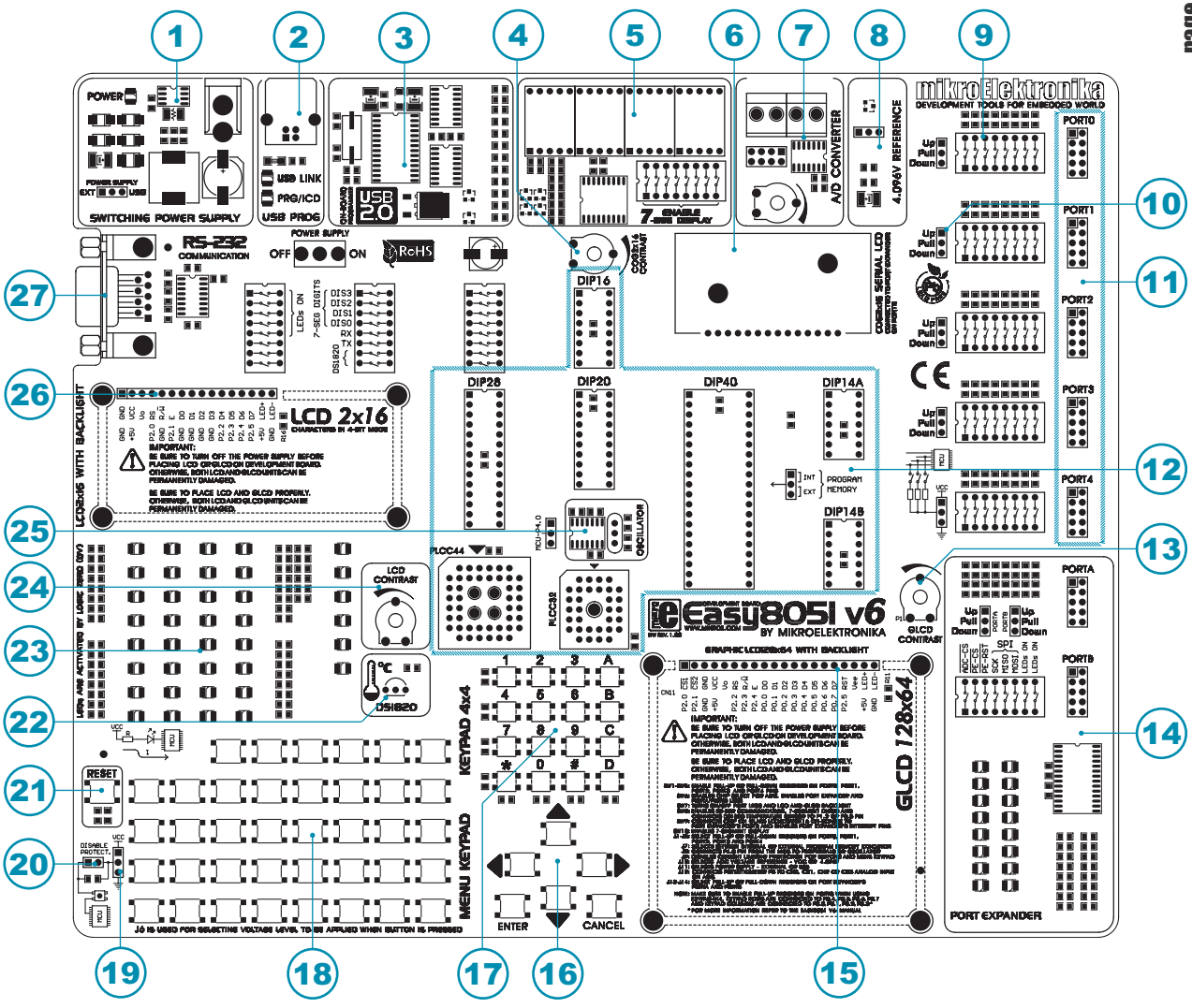
The *8051flash*™ program provides a complete list of all supported microcontrollers. The latest version of this program with updated list of supported microcontrollers can be downloaded from our website at **www.mikroe.com**

Package contains:

Development system:	Easy8051 v6
CD:	product CD with appropriate software
Cables:	USB cable
Documentation:	Easy8051 v6 and 8051flash manuals, Installing USB drivers quick guide and electrical schematic of the Easy8051 v6 development system

System specification:

Power supply:	over a DC connector (7 to 23V AC or 9 to 32V DC); or over a USB cable for programming (5V DC)
Power consumption:	50mA in idle state (when on-board modules are inactive)
Dimension:	26,5 x 22cm (10,4 x 8,6inch)
Weight:	~417g (0.92lbs)



Key Features

1. Power supply voltage regulator
2. On-board programmer's USB connector
3. USB 2.0 *8051prog* programmer
4. On-board 2x16 LCD display contrast adjustment
5. Seven-segment display
6. On-board alphanumeric 2x16 LCD display
7. A/D converter test inputs
8. 4.096V voltage reference
9. DIP switches to enable pull-up/pull-down resistors
10. Pull-up/pull-down resistor selection
11. I/O port connectors
12. Sockets to plug 8051 microcontrollers into
13. Graphic LCD display contrast adjustment
14. Port expander
15. Graphic LCD display connector
16. MENU keypad
17. 4x4 keypad
18. Push buttons to simulate digital inputs
19. Pins' logic state selector
20. Jumper for shortening protective resistor
21. Reset button
22. DS1820 temperature sensor
23. 38 LEDs to indicate pins' logic state
24. Alphanumeric LCD display contrast adjustment
25. Clock signal generator
26. Alphanumeric LCD display connector
27. RS-232 communication connector

1.0. Connecting the System to your PC

Step 1:

Follow the instructions provided in the relevant manuals and install the *8051flash* program and USB drivers from the product CD. USB drivers are necessary for the proper operation of the on-board programmer.
In case you already have some of the Mikroelektronika's compilers installed on your PC, there is no need to reinstall drivers as they will automatically be installed along with the compiler.

Step 2:

Use the USB cable to connect the *Easy8051 v6* development system to your PC. One end of the USB cable with a connector of the USB **B** type should be connected to the development system, as shown in Figure 1-2, whereas the other end of the cable (USB **A** type) should be connected to your PC. When establishing a connection, make sure that jumper J11 is placed in the USB position as shown in Figure 1-1.

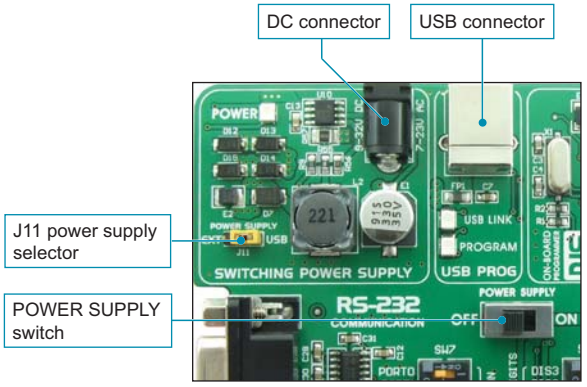


Figure 1-1: Power supply

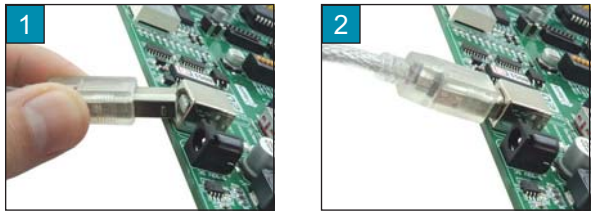


Figure 1-2: Connecting USB cable

Step 3:

Turn on your development system by setting the POWER SUPPLY switch to the ON position. Two LEDs referred to by *POWER* and *USB LINK* will illuminate to indicate that your development system is ready to use. Use the on-board *8051prog* programmer and *8051flash* program to dump a code into the microcontroller and employ the board to test and develop your projects.

NOTE: If you use additional modules, such as LCD, GLCD etc., it is necessary to place them properly on the development board before it is turned on. Otherwise, both additional modules and development system can be permanently damaged. Refer to Figure 1-3 for their proper placing.

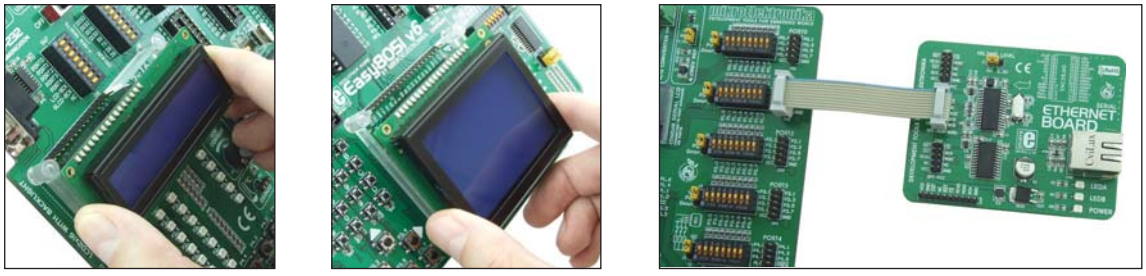


Figure 1-3: Placing additional modules on the board

2.0. Supported Microcontrollers

The *Easy8051 v6* development system provides eight separate sockets for 8051 microcontrollers in DIP40, DIP28, DIP20, DIP16, DIP14A, DIP14B, PLCC44 and PLCC32 packages. These sockets allow supported microcontrollers in DIP packages to be directly plugged into the development board.

Figure 2-1: Microcontroller sockets

There are two sockets for 8051 microcontrollers in DIP14 package provided on the board. Which of these sockets will you use depends solely on the pinout of the microcontroller in use. The *Easy8051 v6* development system comes with the microcontroller in DIP40 package.
Jumpers J7, J8 and J16 next to the DIP40 socket and oscillator are used for selecting functions of the microcontroller pins.

Jumper	Position	Function
J7	INT	Internal memory in use
	EXT	Extenal memory in use
J8	OSC	Pin P4.0 is supplied with a clock signal from the external oscillator
	P4.0	P4.0 is an I/O pin
J16	OSC	Pin P3.2 is supplied with a clock signal from the external oscillator
	P3.2	P3.2 is an I/O pin

8051 microcontrollers can use either internal (on-board) or external oscillator for the generation of the clock signals. There is a quartz oscillator used as an external oscillator provided on the board. Signals generated by such an oscillator are used as clock signals for most supported microcontrollers.

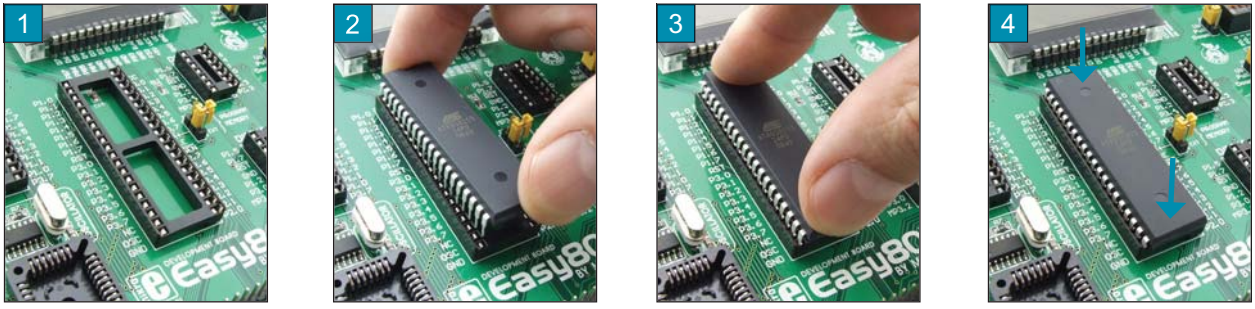


Figure 2-2: Plugging a microcontroller into appropriate socket

Prior to plugging a microcontroller into the appropriate socket, make sure that the power supply is turned off. It is shown in Figure 2-2 how to correctly plug in a microcontroller. Figure 1 shows an unoccupied DIP40 socket. Place one end of the microcontroller into the socket as shown in Figure 2. Then put the microcontroller slowly down until all the pins thereof match the socket as shown in Figure 3. Check again that everything is placed correctly and press the microcontroller down easily until it is completely plugged into the socket, as shown in Figure 4.

NOTE: Only one microcontroller may be placed on the development board at the same time.

page

3.0. On-board USB 2.0 8051prog Programmer

A programmer is a necessary tool when working with microcontrollers as it is used for loading a HEX code into the microcontroller. The *Easy8051 v6* development system includes an on-board *8051prog* programmer. All you need is a .hex file to be loaded into the microcontroller using the *8051flash* programmer. Figure 3-2 shows the connection between the compiler, *8051flash* program and microcontroller.

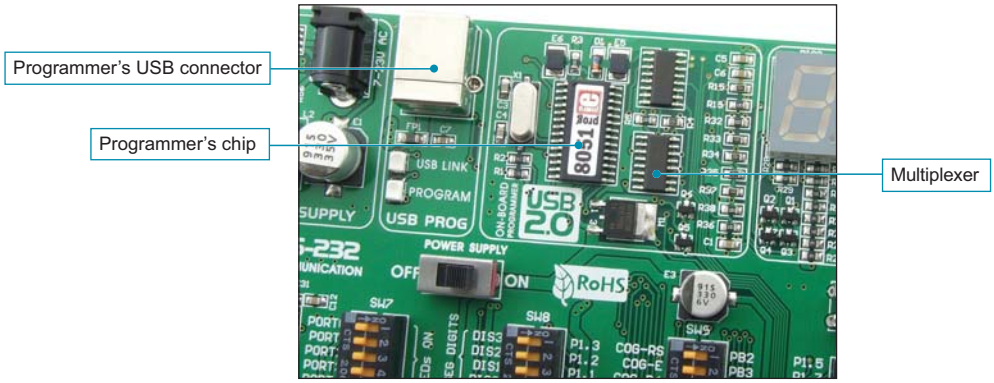


Figure 3-1: *8051prog* programmer

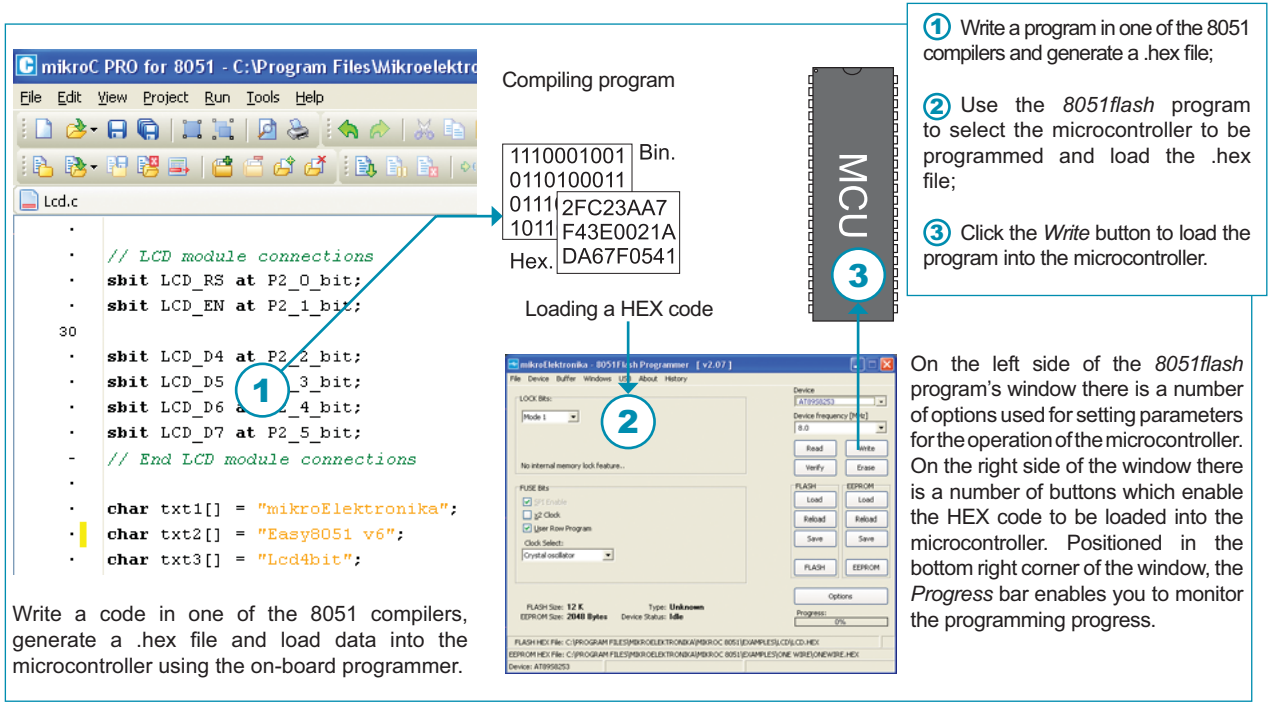
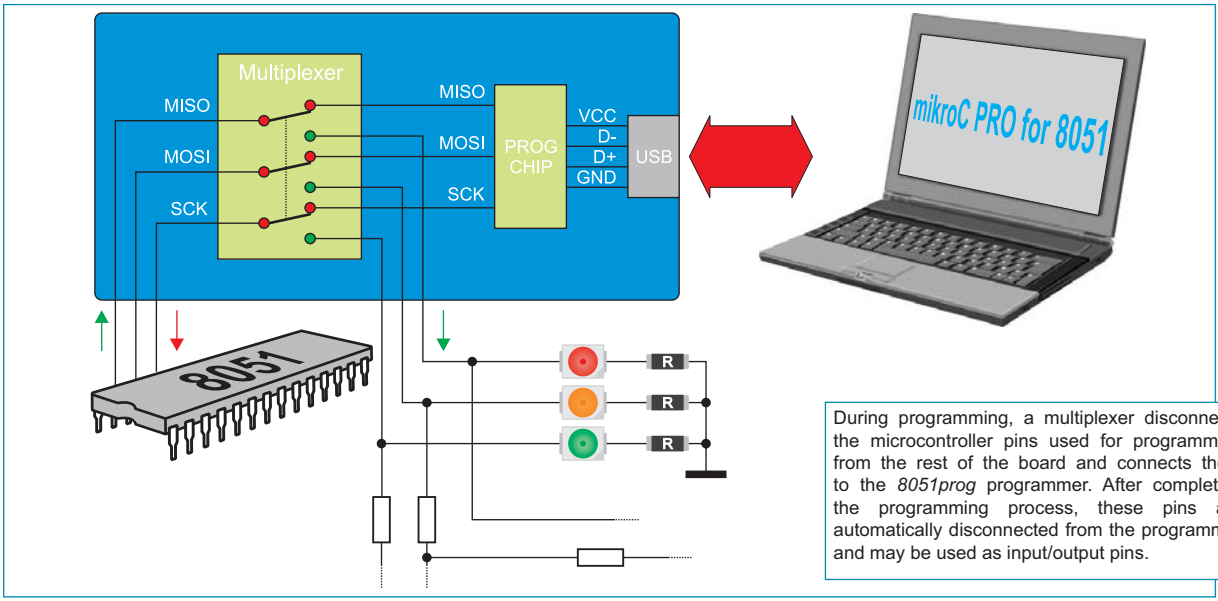

mikroC PRO for 8051 - C:\Program Files\Mikroelektro
File Edit View Project Run Tools Help
Lcd.c
// LCD module connections
sbit LCD_RS at P2_0_bit;
sbit LCD_EN at P2_1_bit;
sbit LCD_D4 at P2_2_bit;
sbit LCD_D5 at P2_3_bit;
sbit LCD_D6 at P2_4_bit;
sbit LCD_D7 at P2_5_bit;
// End LCD module connections
char txt1[] = "mikroElektronika";
char txt2[] = "Easy8051 v6";
char txt3[] = "Lcd4bit";
Compiling program
1110001001 Bin.
0110100011
0111 2FC23AA7
1011 F43E0021A
Hex. DA67F0541
Loading a HEX code
MCU
1
2
3
1 Write a program in one of the 8051 compilers and generate a .hex file;
2 Use the 8051flash program to select the microcontroller to be programmed and load the .hex file;
3 Click the Write button to load the program into the microcontroller.
Write a code in one of the 8051 compilers, generate a .hex file and load data into the microcontroller using the on-board programmer.
On the left side of the 8051flash program's window there is a number of options used for setting parameters for the operation of the microcontroller. On the right side of the window there is a number of buttons which enable the HEX code to be loaded into the microcontroller. Positioned in the bottom right corner of the window, the Progress bar enables you to monitor the programming progress.

Figure 3-2: The principle of the programmer's operation

NOTE: For more information on the *8051prog* programmer refer to the relevant manual provided in the *Easy8051 v6* development system package.

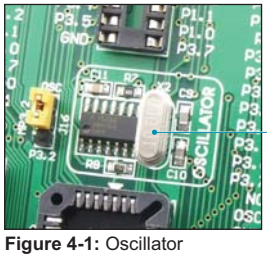
8051 microcontrollers are programmed using SPI serial communication which employs MISO, MOSI and SCK microcontroller pins.



During programming, a multiplexer disconnects the microcontroller pins used for programming from the rest of the board and connects them to the *8051prog* programmer. After completing the programming process, these pins are automatically disconnected from the programmer and may be used as input/output pins.

4.0. Clock Oscillator

There is a quartz oscillator used as an external clock signal source provided on the board. A quartz crystal, used for stabilizing clock frequency, is plugged into the appropriate socket and can always be replaced with another one. The maximum value thereof depends on the microcontroller maximum operating frequency allowed. The function of the P4.0/OSC pin depends on the jumper J8 position. When this jumper is set to the OSC position, the P4.0/OSC pin will be fed with a clock signal. When it is set to the P4.0 position, the P4.0/OSC pin is available as input/output pin. The same is with jumper J16 and the MP3.2 pin. When this jumper is set to the OSC position, the MP3.2 pin will be fed with a clock signal. When it is set to the P3.2 position, the MP3.2 pin is available as an input/output pin.



Quartz-crystal X2 is placed in appropriate socket and can be easily replaced with another one.

Figure 4-1: Oscillator

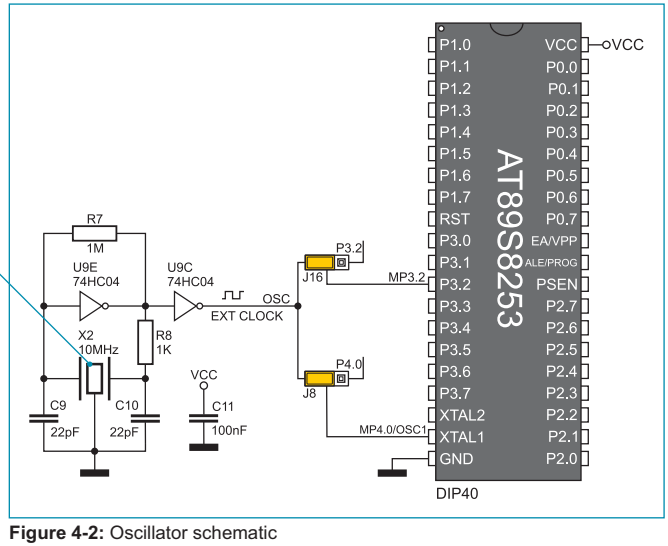


Figure 4-2: Oscillator schematic

page

5.0. Power Supply

The *Easy8051 v6* development system may use either of the two following power supply sources:

1. +5V PC power supply through the USB programming cable;
2. External power supply connected to a DC connector provided on the development board.

The MC34063A voltage regulator and Gretz rectifier are used for enabling external power supply voltage to be either AC (in the range of 7V to 23V) or DC (in the range of 9V to 32V). Jumper J11 is used as a selector for a power supply source. In order to use the USB power supply it is necessary to place jumper J11 in the USB position. When using external power supply, jumper J11 should be placed in the EXT position. The development system is turned on/off by setting the POWER SUPPLY switch to the ON or OFF position, respectively.

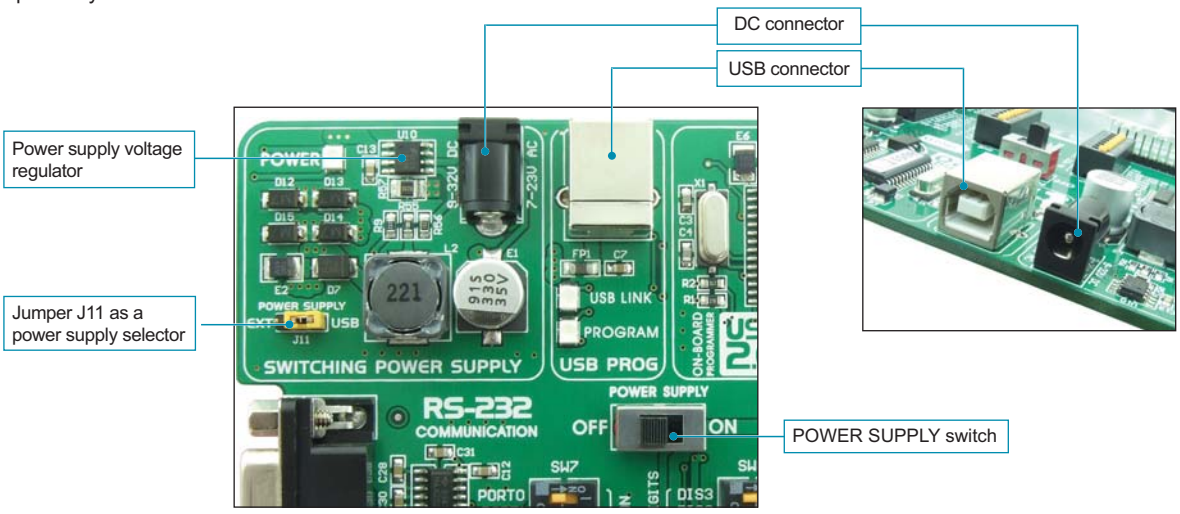


Figure 5-1: Power supply

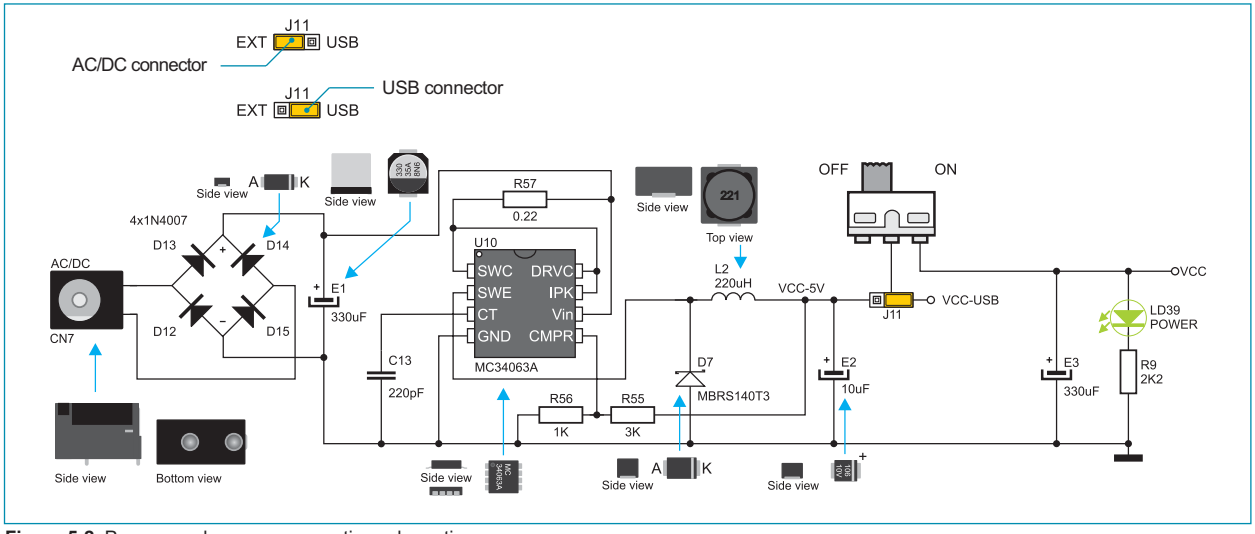


Figure 5-2: Power supply source connection schematic

6.0. RS-232 Communication Interface

The USART (universal synchronous/asynchronous receiver/transmitter) is one of the most common ways of exchanging data between the PC and peripheral components. RS-232 serial communication is performed through a 9-pin SUB-D connector and the microcontroller USART module. The *Easy8051 v6* development system provides one RS-232 port which is enabled by setting swiches 5 and 6 on the DIP switch SW8 to the ON position. The microcontroller pins used in such communication are marked as follows: RX - *receive data* and TX - *transmit data*. Baud rate goes up to 115 kbps.

In order to enable the microcontroller's USART module to receive input signals with different voltage levels, it is necessary to provide a voltage level converter such as MAX202C (MAX232).

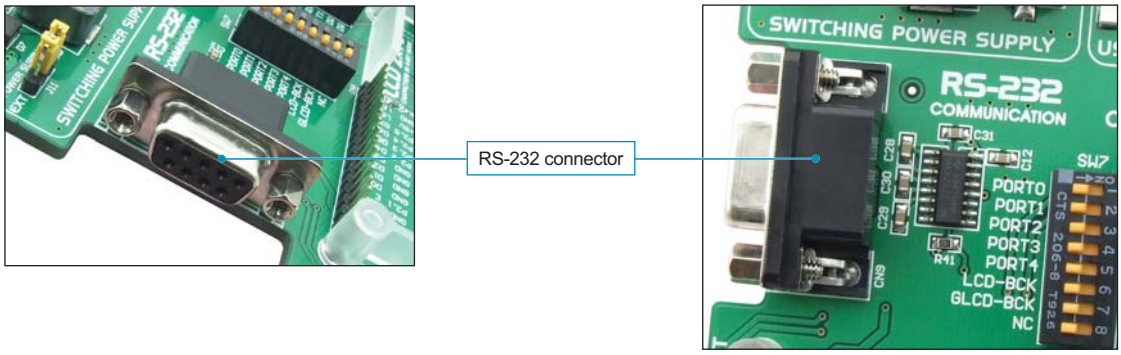


Figure 6-1: RS-232 module

Switches 5 and 6 on the DIP switch SW8 are used to interface the RS-232 module to the microcontroller via RX and TX lines, as shown in Figure 6-2.

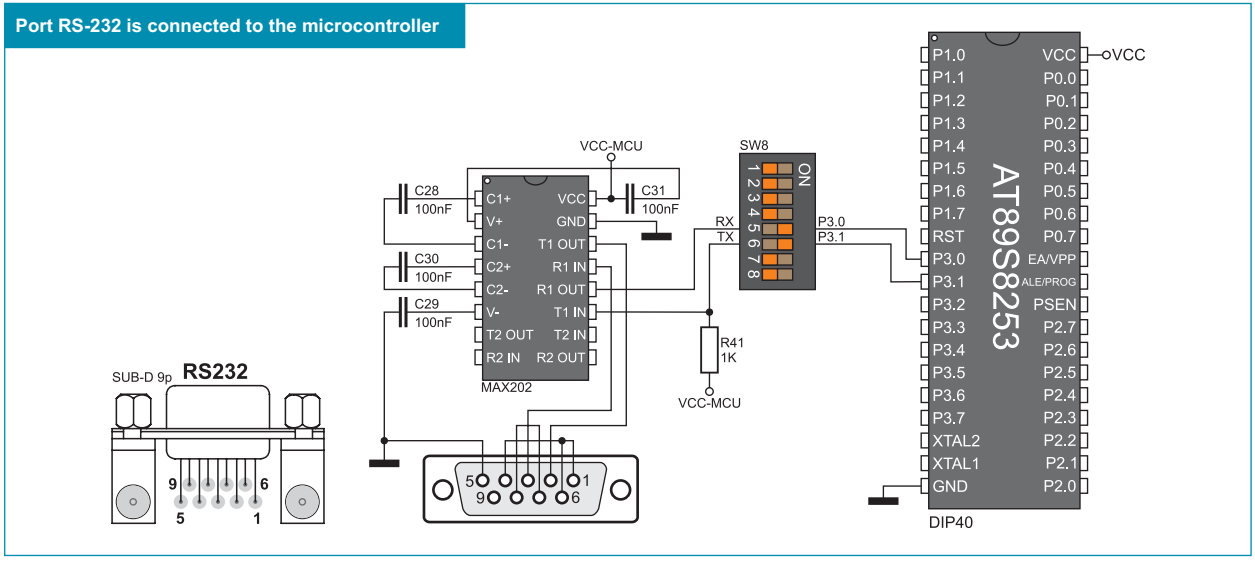


Figure 6-2: RS-232 module schematic

NOTE: Make sure that your microcontroller is provided with the USART module as it is not necessarily integrated in all PIC microcontrollers.

page

7.0. A/D Converter

An A/D converter is used for converting an analog signal into the appropriate digital value. A/D converter is linear, which means that the converted number is linearly dependent on the input voltage value. The *Easy8051 v6* development uses the MCP3204 circuit as an A/D converter. Voltage to be converted is brought to the input pins of this circuit which then converts it into a 12-bit digital value. Such number is transferred to the microcontroller via serial communication, while switches 1, 4, 5 and 6 on the DIP switch SW6 should be set to the ON position. Voltage provided on the A/D converter input pins can be either power supply voltage (5V) or external voltage brought to the converter via CN12 and CN13 connectors. When the power supply voltage is used, it is necessary to determine which of the following pins CH0, CH1, CH2 or CH3 is to be supplied with such voltage using jumper J12. In this case, the voltage value on the A/D converter input can vary between 0V and 5V using potentiometer P3. When the external voltage is used, it is necessary to remove jumpers J12 in order to prevent voltage interference and provide conversion accuracy. The MCP3204 circuit includes the Vref pin to receive voltage reference. However, the main purpose of this pin is to determine the range of analog voltage to be converted, hence the A/D conversion resolution. Jumper J10 is used for selecting one of two possible voltage reference sources. When this jumper is in the VCC position, the value of the voltage reference is 5V, whereas it amounts to 4.096V when the jumper is set to the position referred to by 4.096. The latter is generated by the MCP1541 circuit.

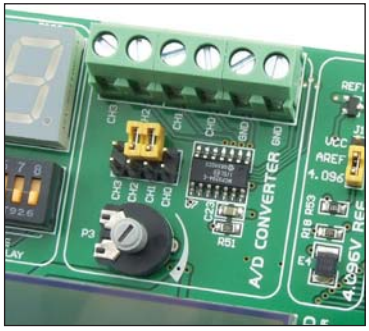


Figure 7-1: ADC (jumper default position)

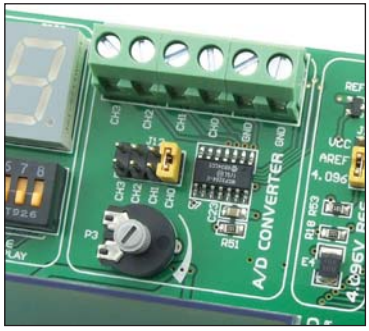


Figure 7-2: Pin CH0 used as A/D conversion input

A/D conversion is performed via the CH0 pin on A/D convertor

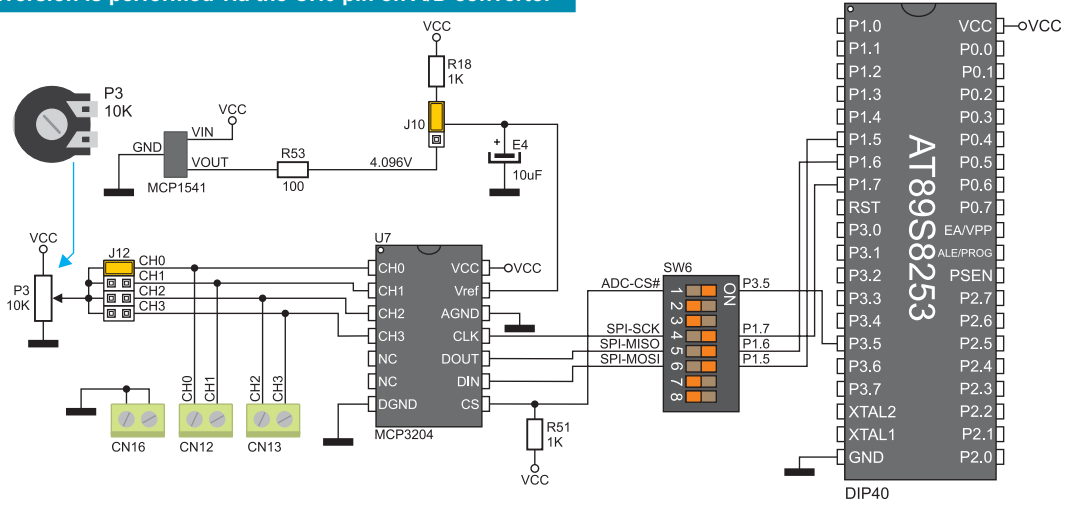


Figure 7-3: Microcontroller and A/D converter test inputs connection schematic

8.0. DS1820 Temperature Sensor

1-wire® serial communication enables data to be transferred over a single communication line while the process itself is under the control of the *master* microcontroller. The advantage of such communication is that only one microcontroller pin is used. All *slave* devices have by default a unique ID code, which enables the master device to easily identify all devices sharing the same interface.

DS1820 is a temperature sensor that uses 1-wire standard for its operation. It is capable of measuring temperatures within the range of -55 to 125°C and provides ±0.5°C accuracy for temperatures within the range of -10 to 85°C. Power supply voltage of 3V to 5.5V is required for its operation. It takes maximum 750ms for the DS1820 to calculate temperature with 9-bit resolution. The *Easy8051 v6* development system provides a separate socket for the DS1820. It may use either P1.2 or P3.3 pin for communication with the microcontroller, which depends on the position of switches 7 and 8 on the DIP switch SW8. In Figure 8-5, switch 7 on the DIP switch SW8 is in the ON position which means that communication is enabled via the P1.2 pin.

Figure 8-1: DS1820 connector (DS1820 is not placed)

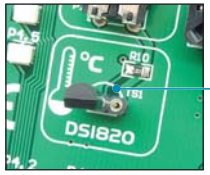


Figure 8-2: DS1820 is plugged into the connector

NOTE:
Make sure that half-circle on the board matches the round side of the DS1820

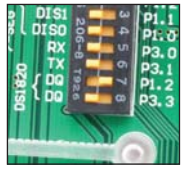


Figure 8-3: Switch 7 on the DIP switch SW8 is in the ON position, DS1820 is connected to the P1.2 pin

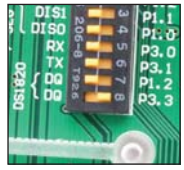


Figure 8-4: Switch 8 on the DIP switch SW8 is in the ON position, DS1820 is connected to the P3.3 pin

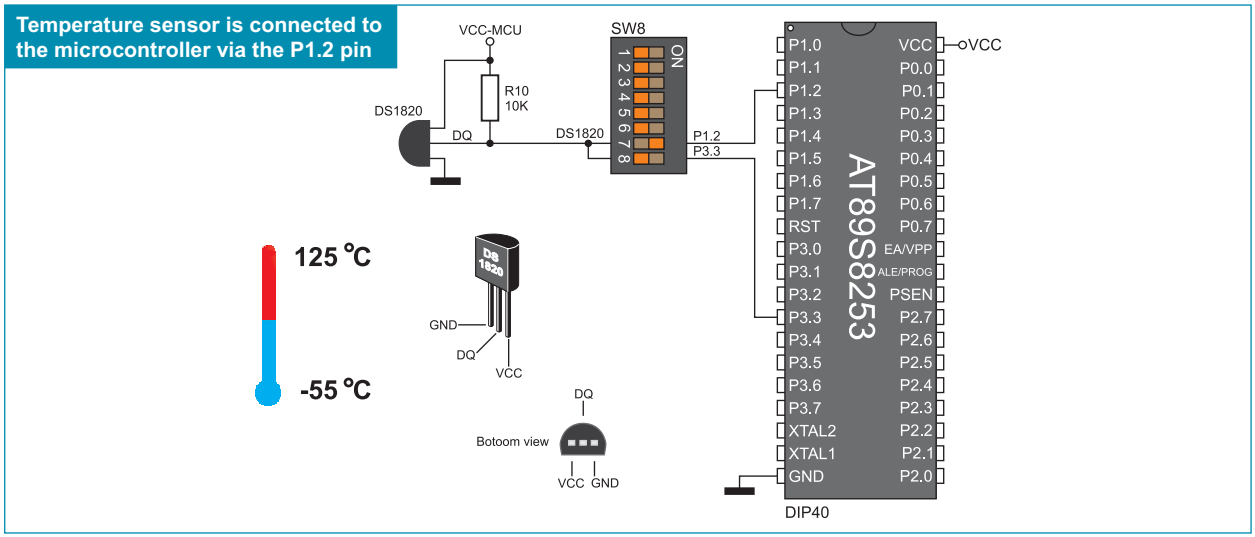


Figure 8-5: DS1820 and microcontroller connection schematic

page

9.0. LEDs

LED diode (Light-Emitting Diode) is a highly efficient electronic light source. When connecting LEDs, it is necessary to use a current limiting resistor. A common LED diode voltage is approximately 2.5V, while the current varies from 1mA to 20mA depending on the type of LED diode. The *Easy8051 v6* development system uses LEDs with current I=1mA.

The *Easy8051 v6* development has 38 LEDs which visually indicate the state of each microcontroller I/O pin. An active LED diode indicates that a logic one (1) is present on the pin. In order to enable the pin state to be shown, it is necessary to select appropriate port PORT0, PORT1, PORT2, PORT3 or PORT4 using the DIP switch SW7.

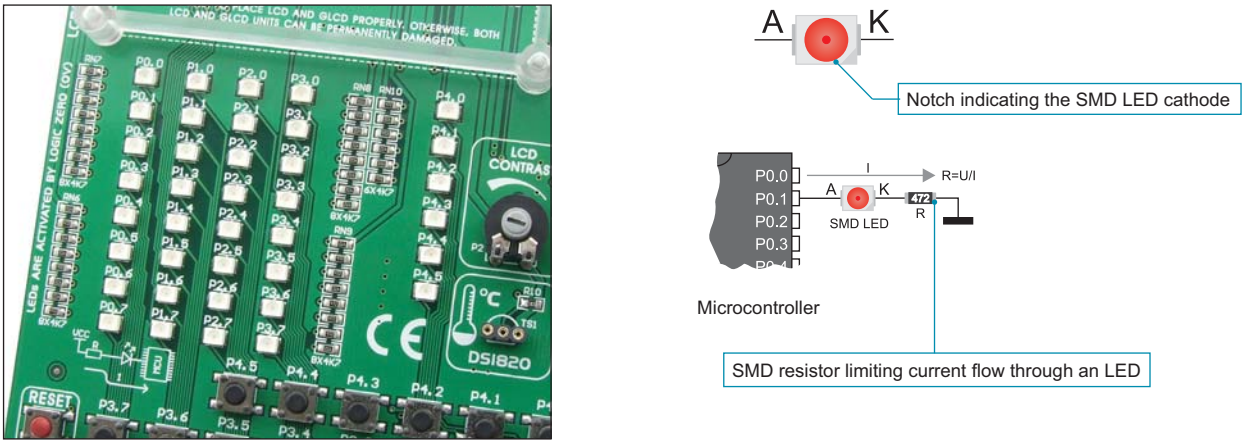


Figure 9-1: LEDs

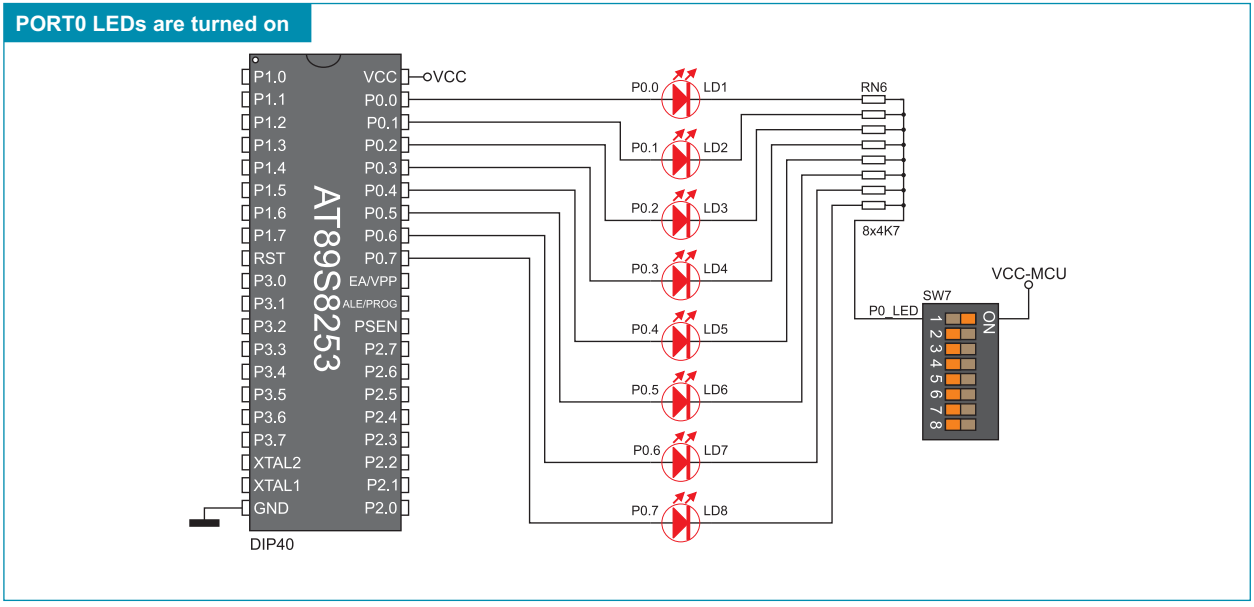


Figure 9-2: LED diode and port PORT0 connection schematic

10.0. Push Buttons

The logic state of all microcontroller digital inputs may be changed using the push buttons. Jumper J6 is used to determine the logic state to be applied to the desired microcontroller pin by pressing the appropriate push button. The purpose of the protective resistor is to limit the maximum current, thus preventing a short circuit from occurring. If needed, advanced users may shorten such resistor using jumper J9. Right next to the push buttons, there is a RESET button which is not connected to the mRST pin. The reset signal is generated by the programmer.

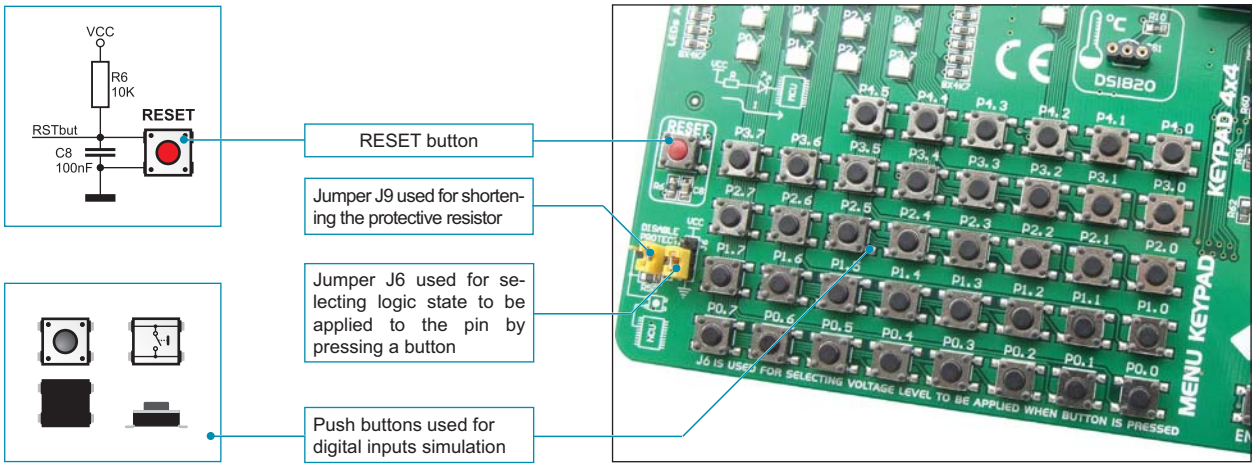

VCC
R6
10K
RESET
RSTbut
C8
100nF
RESET button
Jumper J9 used for shortening the protective resistor
Jumper J6 used for selecting logic state to be applied to the pin by pressing a button
Push buttons used for digital inputs simulation
DS1820
KEYPAD 4x4
MENU KEYPAD
J6 IS USED FOR SELECTING VOLTAGE LEVEL TO BE APPLIED WHEN BUTTON IS PRESSED

Figure 10-1: Push buttons

Pressure on any push button, when jumper J6 is in the VCC position, will cause a logic one (5V) to be applied to appropriate microcontroller pin as shown in Figure 10-2.

Jumper J6 in the VCC position

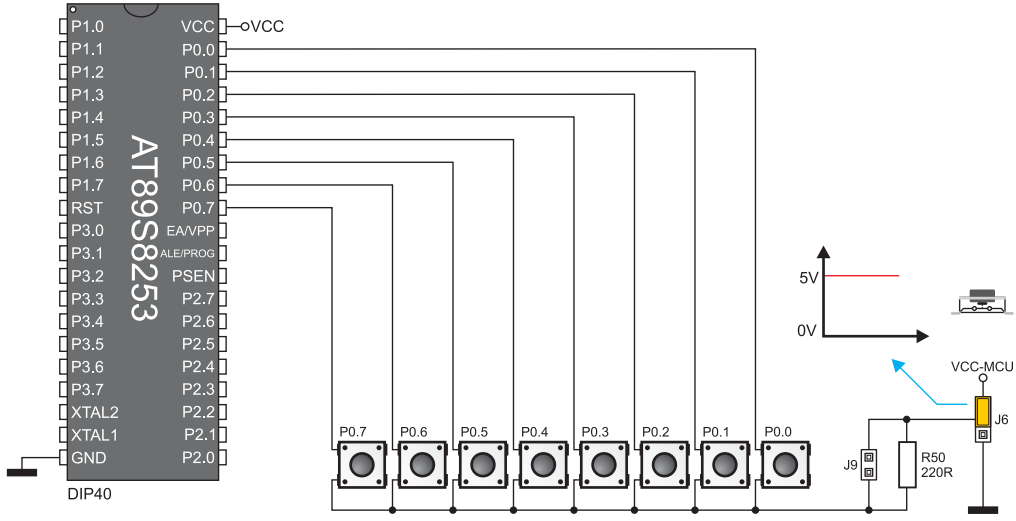


Figure 10-2: Push buttons and port PORT0 connection schematic

page

11.0. MENU Keypad

There is a group of push buttons on the *Easy8051 v6* development system which constitute a navigation keypad called MENU. The main feature of the MENU keypad is the arrangement of its push buttons marked as left, right, up and down arrows. Besides, there are also two additional push buttons referred to by ENTER and CANCEL. MENU push buttons are connected in the same way as the port PORT3 push buttons. Their function is determined by the user when writing the program for the microcontroller.

Have in mind when writing a program for the microcontroller that the MENU keypad is connected to the PORT3 port

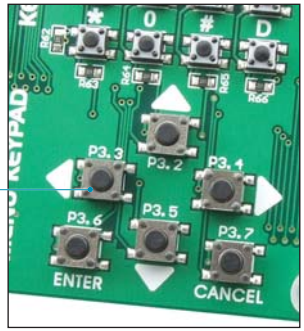


Figure 11-1: MENU keypad

MENU keypad push buttons are connected in the same manner as port PORT3 push buttons

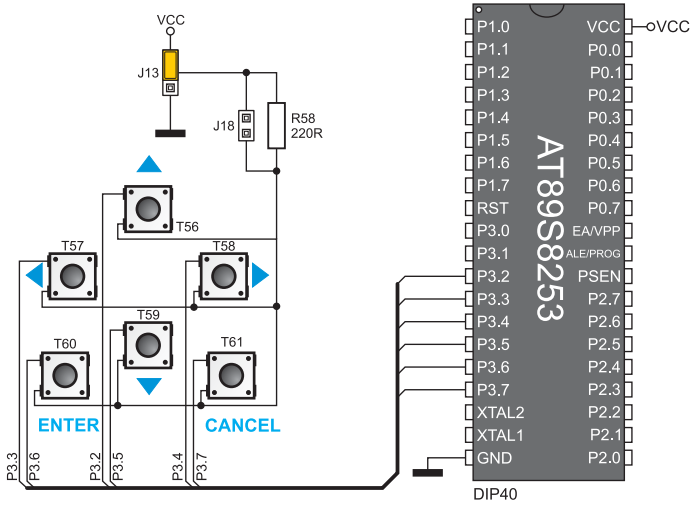


Figure 11-2: MENU keypad and microcontroller connection schematic

12.0. 4x4 Keypad

The 4x4 keypad is a standard alphanumeric keypad connected to the microcontroller PORT0. The operation of such keypad is based on the 'scan and sense' principle where the P0.4, P0.5, P0.6 and P0.7 pins are configured as inputs connected to pull-down resistors. Pins P0.0, P0.1, P0.2 and P0.3 are configured as high level voltage outputs. Pressure on any button will cause a logic one (1) to be applied to one of input pins. It is determined which of the push buttons is pressed from within the software. For example, by pressing button 6, a logic one (1) will appear on the P0.5 pin. After that, by applying a logic one (1) to each of the following output pins P0.0, P0.1, P0.2 and P0.3 it is possible to determine which of the push buttons is pressed.

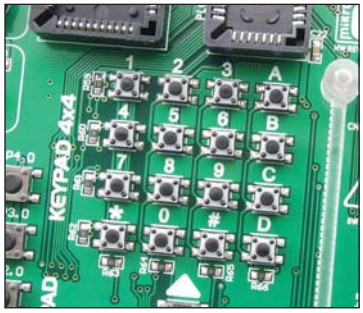


Figure 12-1: 4x4 keypad

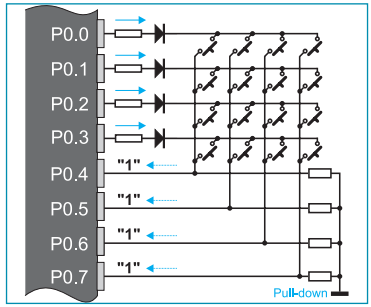


Figure 12-2: 4x4 keypad performance

Jumper J1 is in the GND position. Pins P0.4, P0.5, P0.6 and P0.7 are connected to pull-down resistors via DIP switch SW1.

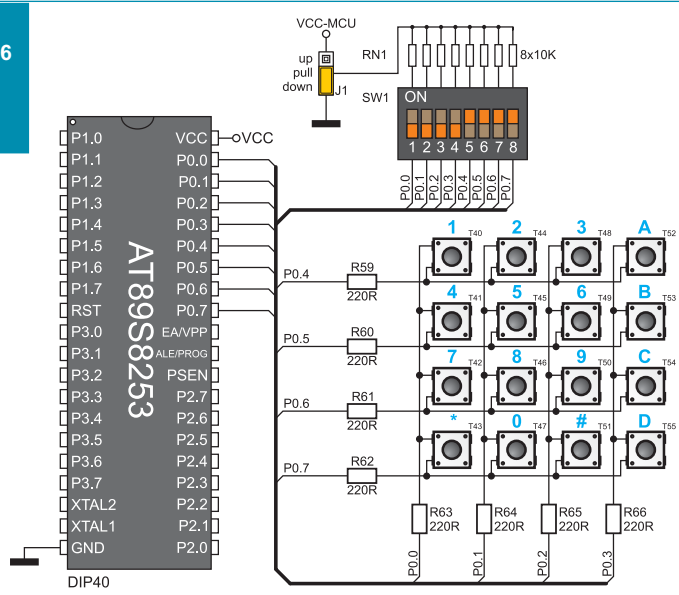


Figure 12-3: 4x4 keypad and microcontroller connection schematic

page

13.0. On-board 2x16 LCD Display

The on-board LCD display is connected to the microcontroller via the port expander. In order to enable such a display to be used it is necessary to set switches 1-6 DIP on the DIP switch SW9 to the ON position, thus connecting the on-board display's pins to port expander's port PORTB. SPI communication between the port expander and microcontroller is enabled by means of the DIP switch SW6. Potentiometer P5 is used for the display contrast adjustment. Unlike additional LCD display, the on-board LCD display has no backlight, whereas both displays shows digits in two lines each containing up to 16 characters of 7x5 pixels.

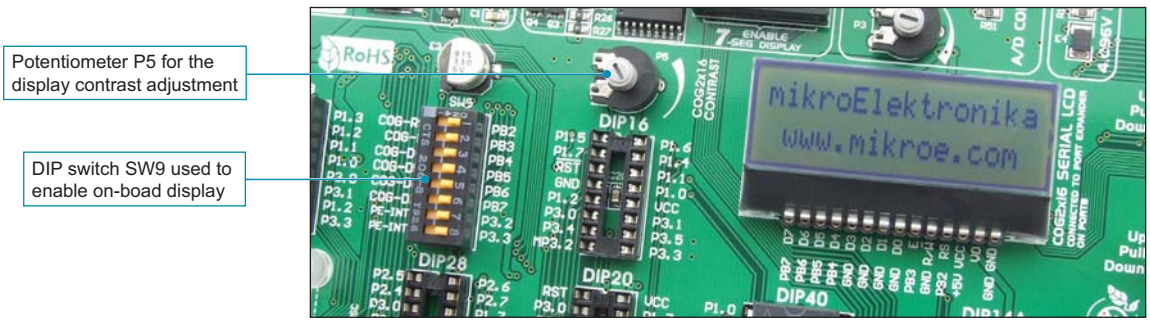


Figure 13-1: On-board 2x16 LCD display

SW6: CS, RST, SCK, MISO, MOSI = ON
SW9: 1-8 = ON

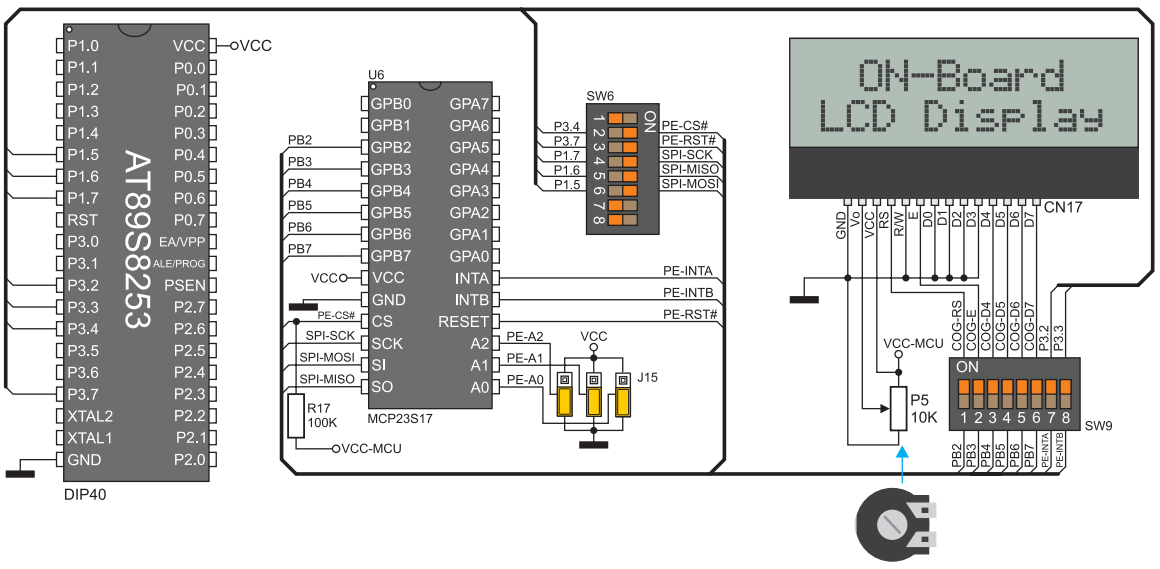


Figure 13-2: On-board LCD display connection schematic

14.0. 2x16 LCD Display

The *Easy8051 v6* development system provides an on-board connector for alphanumeric 2x16 LCD display to be plugged into. Such connector is linked to the microcontroller through the PORT2 port. Potentiometer P2 is used for display contrast adjustment. The switch marked as LCD-BCK on the DIP switch SW7 is used to turn on/off the display backlight.
Communication between the LCD display and the microcontroller is performed in a 4-bit mode. Alphanumeric digits are displayed in two lines each containing up to 16 characters of 7x5 pixels.

Figure 14-1: Alphanumeric 2x16 LCD display connector

Figure 14-2: Alphanumeric 2x16 LCD display

LCD display backlight is turned on

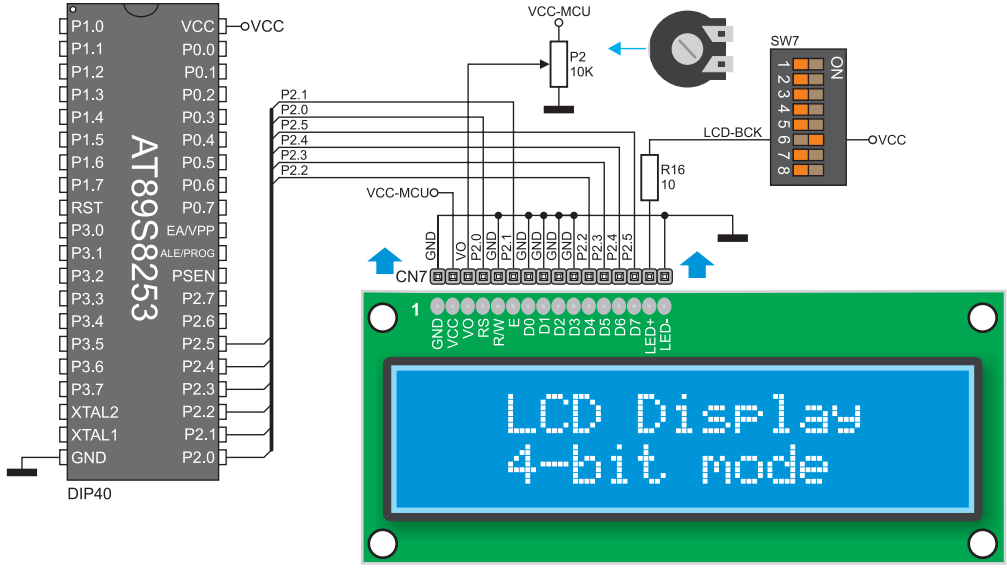


Figure 14-3: Alphanumeric 2x16 LCD display connection schematic

page

15.0. 128x64 Graphic LCD Display

128x64 graphic LCD display (128x64 GLCD) provides an advanced method for displaying graphic messages. It is connected to the microcontroller through PORT0 and PORT2. GLCD display has the screen resolution of 128x64 pixels which allows you to display diagrams, tables and other graphic content. Since the PORT2 port is also used by 2x16 alphanumeric LCD display, you cannot use both displays simultaneously. Potentiometer P1 is used for the GLCD display contrast adjustment. Switch 7 (GLCD-BCK) on the DIP switch SW7 is used to turn the display backlight on/off.

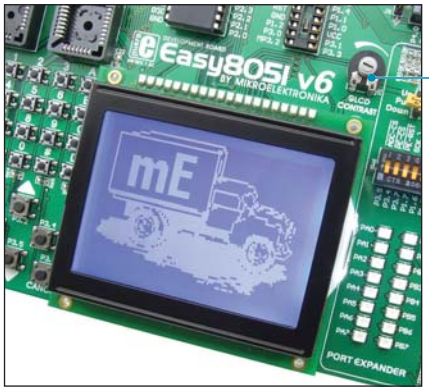


Figure 15-1: GLCD display

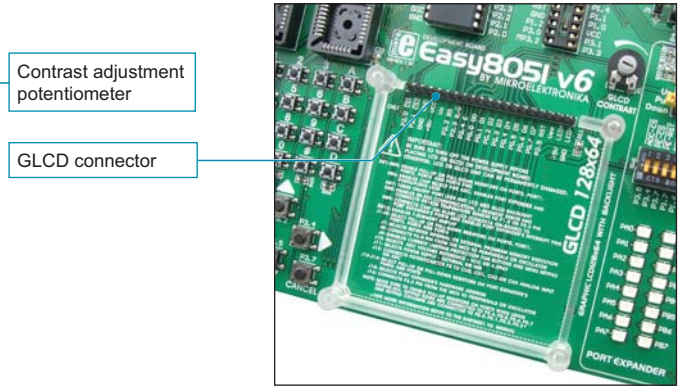


Figure 15-2: GLCD connector

GLCD display backlight is turned on

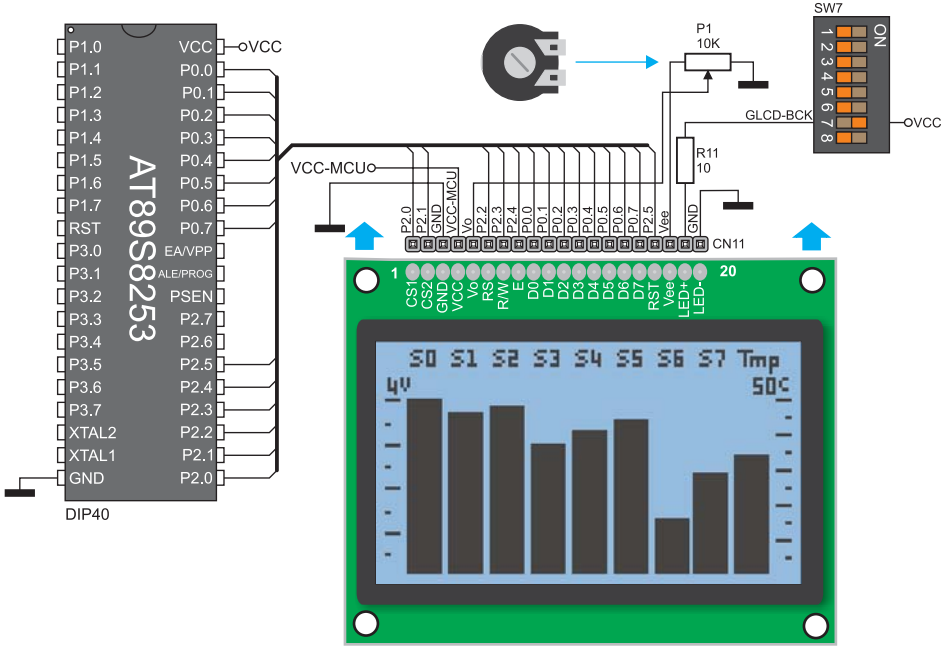


Figure 15-3: GLCD display connection schematic

16.0. Seven-Segment Display

The *Easy8051 v6* development system also includes a seven-segment display with four digits that uses the PORT0 and PORT1 microcontroller ports for its operation. Between the PORT0 port and seven-segment display there is a ULN2803 circuit used for running the segments of the seven-segment display. The operation of such display is enabled by multiplexing, which means that the operation of all four digits requires only eight lines connected to display segments and four lines for the activation of each digit. The PORT0 port is used to send a signal to digit segments, whereas PORT1 specifies a digit to be fed with such signal. In order to enable the operation of the seven-segment display it is necessary to set all switches on the DIP switch SW10 as well as switches 1-4 on the DIP switch SW8 to the ON position.

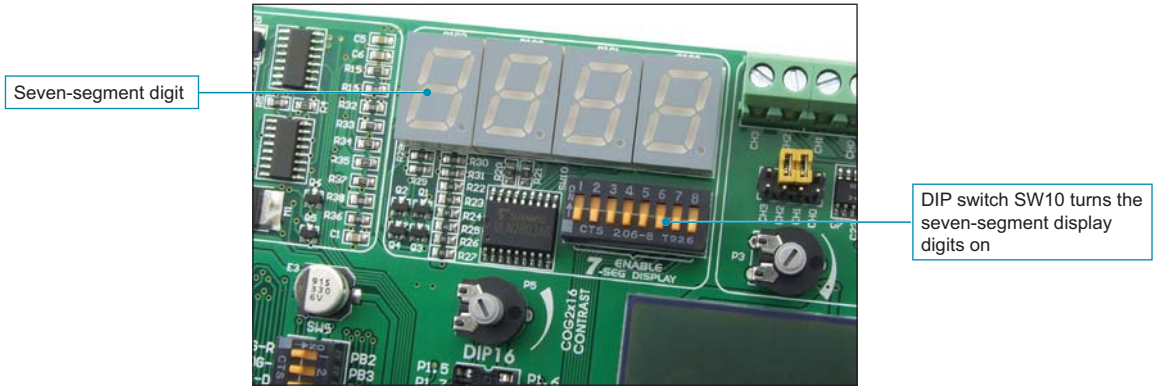


Figure 16-1: GLCD display

The seven-segment display's time multiplexing is based on fast turning display segments on/off in such a manner that one gets impression that all digits are active simultaneously. The microcontroller feeds display segments with signals via the PORT0 port, whereas the PORT1 port selects one of them to display appropriate number. In the example shown in Figure 16-2, signals from the microcontroller cause the DIS3 display segments to show number 5. When such display is off, the microcontroller sends signal to the next digit (DIS2), via the P1.2 pin, to show number 6. The same procedure applies to digits DIS1 and DIS0. When one cycle is complete, another one automatically starts and digits successively change on dislays DIS3, DIS2, DIS1, DIS0, DIS3, DIS2 etc.

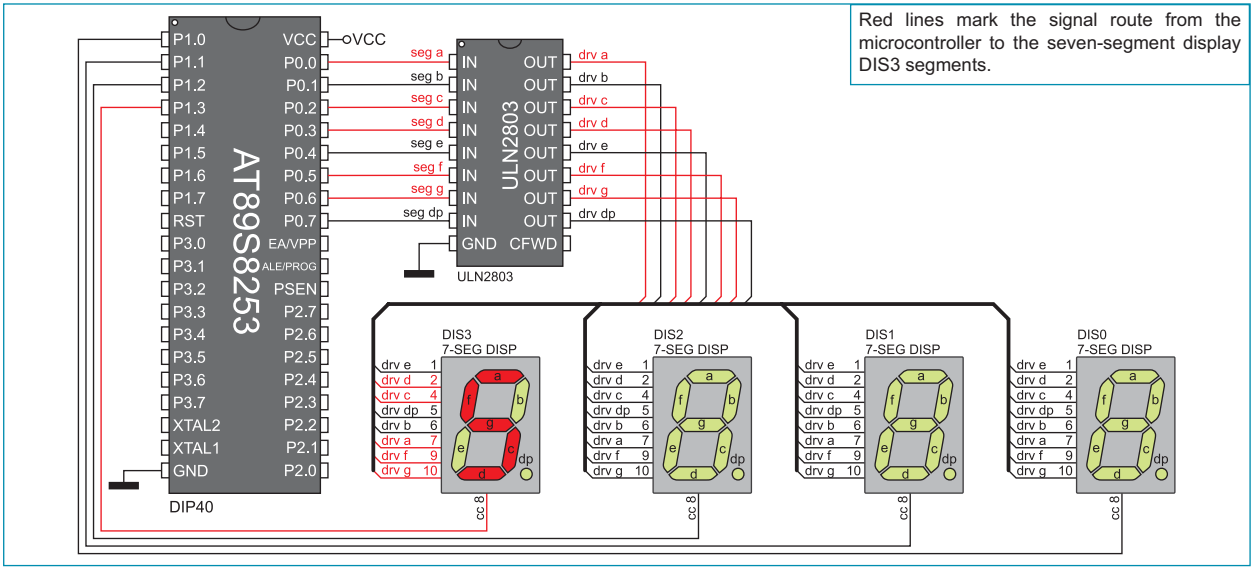


Figure 16-2: Showing a digit on the seven-segment display DIS3

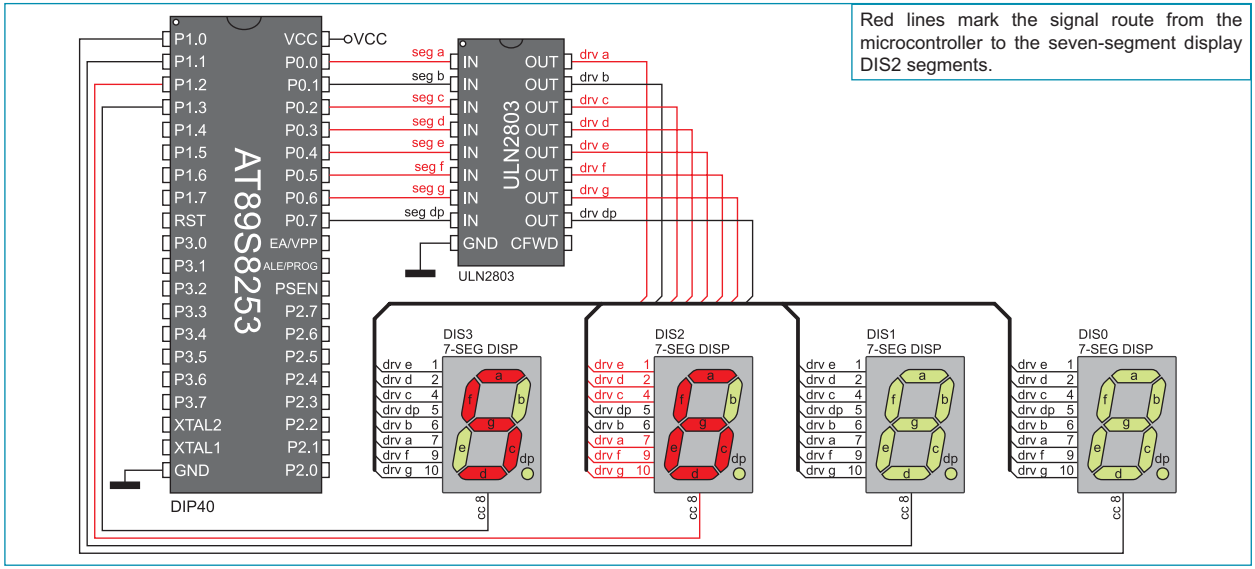


Figure 16-3: Showing a digit on the seven-segment display DIS2

Seven-segment display is enabled

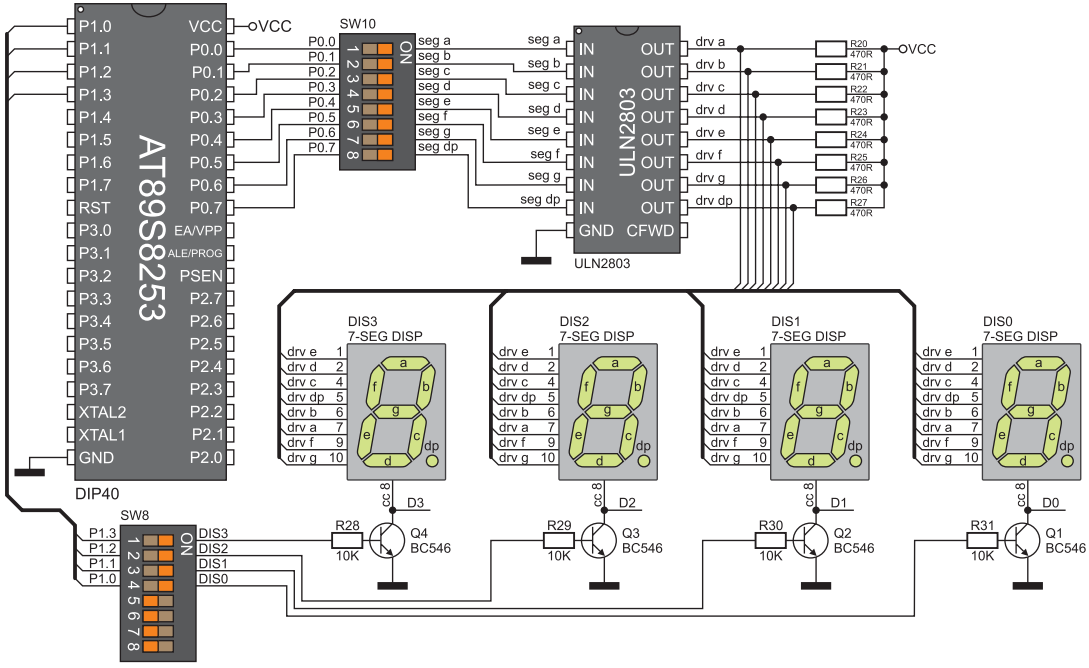


Figure 16-4: Seven-segment display connection schematic

17.0. Input/Output Ports

Along the right side of the development system, there are nine 10-pin connectors which are connected to the microcontroller I/O ports. Pins P1.5, Pi.6 and P1.7 are not directly connected to the appropriate 10-pin connector, but via programmer's multiplexer. DIP switches SW1-SW5 enable each connector pin to be connected to one pull-up/pull-down resistor. Whether port pins are to be connected to a pull-up or pull-down resistor depends on the position of jumpers J1-J5.

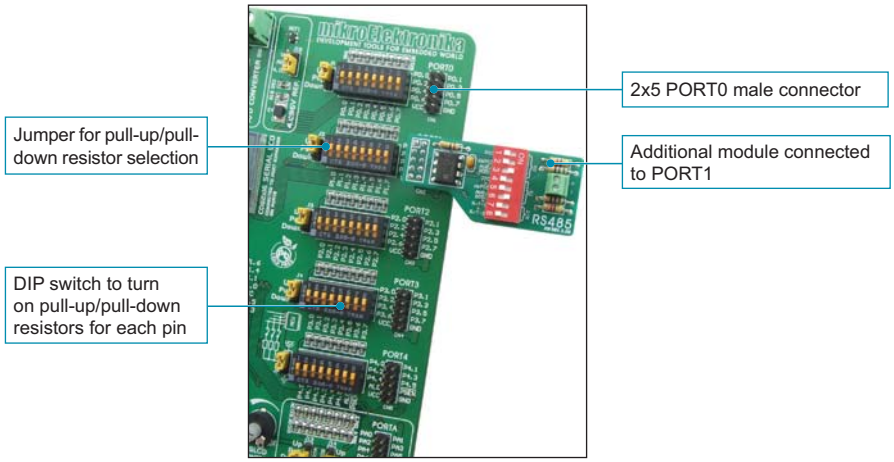

2x5 PORT0 male connector
Jumper for pull-up/pull-down resistor selection
Additional module connected to PORT1
DIP switch to turn on pull-up/pull-down resistors for each pin

Figure 17-1: I/O ports

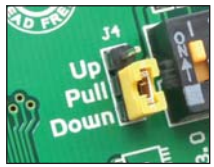


Figure 17-2: J2 in the pull-down positon

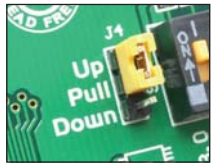


Figure 17-3: J2 in the pull-up position

Microcontroller port PORT0 pins are connected to pull-down resistors

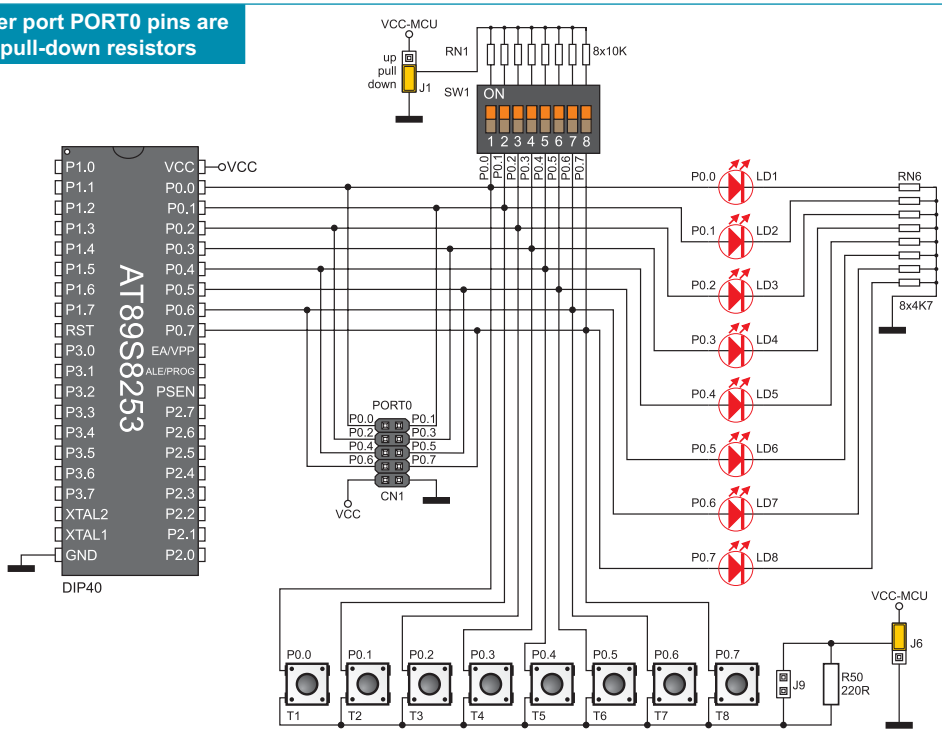


Figure 17-4: Port PORT0 connection schematic

page

Pull-up/pull-down resistors enable you to set the logic level on all microcontroller input pins when they are in idle state. Such level depends on the position of the pull-up/pull-down jumper. The microcontroller pin P0.0 with the relevant DIP switch SW1, jumper J1 and push button P0.0 with jumper J6 are used here for the purpose of explaining the performance of pull-up/pull-down resistors. The principle of their operation is identical for all the microcontroller pins.

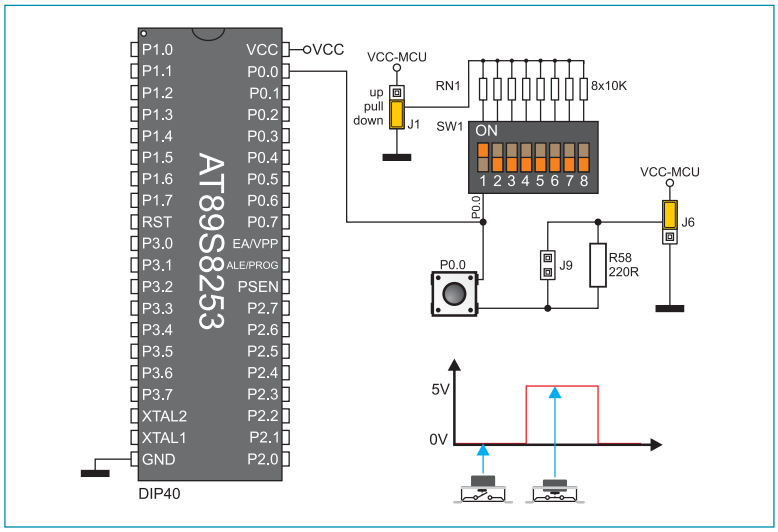


Figure 17-5: Jumper J1 in pull-down and jumper J6 in pull-up position

In order to enable the PORT0 port to be connected to the pull-down resistors, first it is necessary to set jumper J1 to the *Down* position. This enables any port PORT0 pin to be provided with a logic zero (0V) in idle state over jumper J1 and 8x10k resistor network. To provide the P0.0 pin with such a signal, it is necessary to set the P0.0 switch on the DIP switch SW1 to the ON position.

As a result, every time you press the P0.0 push button, a logic one (VCC) will appear on the P0.0 pin, provided that jumper J6 is set to the position referred to by VCC.

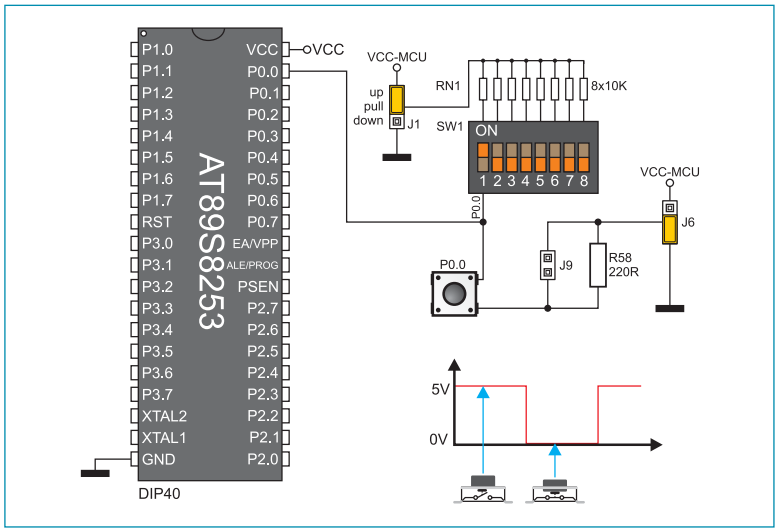


Figure 17-6: Jumper J1 in pull-up and jumper J6 in pull-down position

In order to enable port PORT0 pins to be connected to pull-up resistors and port input pins to be activated with logic zero (0), it is necessary to set jumper J1 in position referred to as *Up* and jumper J6 in position referred to as GND. This enables any port PORT0 input pin to be provided with a logic one (5V) in idle state over the 10k resistor. The P0.0 switch should be set to the ON position afterwards.

As a result, every time you press the P0.0 push button, a logic zero (0) will apear on the P0.0 pin.

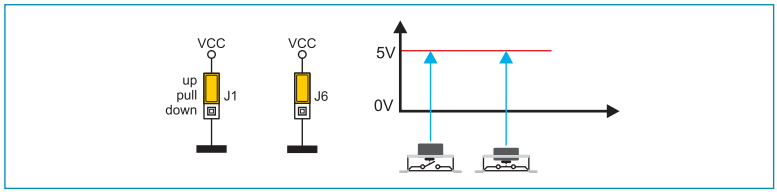


Figure 17-7: Jumpers J1 and J6 in the same position

In case jumpers J1 and J6 have the same logic state, pressure on any button will not cause input pins to change their logic state.

18.0. Port Expander (Additional Input/Output Ports)

The SPI communication lines and MCP23S17 circuit provide the *Easy8051 v6* development system with a means of increasing the number of available I/O ports by two. If the port expander communicates to the microcontroller over the DIP switch SW6 then the microcontroller pins P3.5, P3.4, P3.7, P1.7, P1.6 and P1.5, used for the operation of port expander, cannot be used as I/O pins.

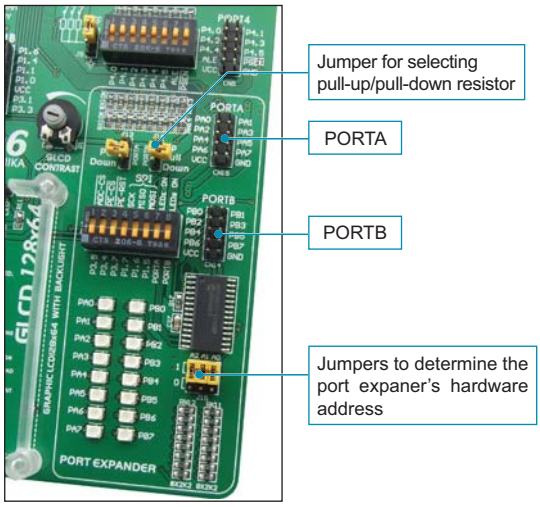


Figure 18-1: Port expander

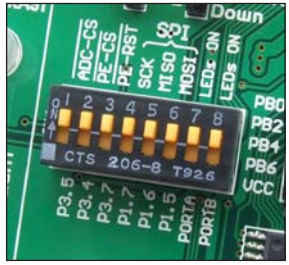


Figure 18-2: Position of DIP switch SW6 when the port expander is enabled

The microcontroller communicates to the port expander (MCP23S17 circuit) using serial communication (SPI). The advantage of such communication is that only five lines are used for transmitting and receiving data simultaneously:

MOSI - Master Output, Slave Input (microcontroller output, MCP23S17 input)
MISO - Master Input, Slave Output (microcontroller input, MCP23S17 output)
SCK - Serial Clock (microcontroller clock signal)
CS - Chip Select (enables data transfer)
RST - Reset

Data transfer is performed in both directions simultaneously by means of MOSI and MISO lines. The MOSI line is used for transferring data from the microcontroller to the port expander, whereas the MISO line transfers data from the port expander to the microcontroller. The microcontroller initializes data transfer when the CS pin is driven low (0V). It causes the microcontroller to send clock signal (SCK) and therefore start data exchange. The principle of operation of the port expander's ports 0 and 1 is almost identical to the operation of other ports on the development system. The only difference here is that the port signals are received in a parallel format. The MCP23S17 then converts such signals into serial format and sends them to the microcontroller. The result is a reduced number of lines used for sending signals from ports 0 and 1 to the microcontroller and vice versa.

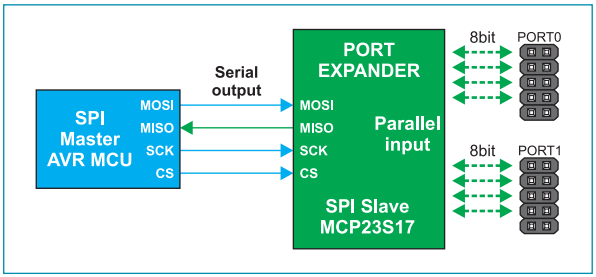
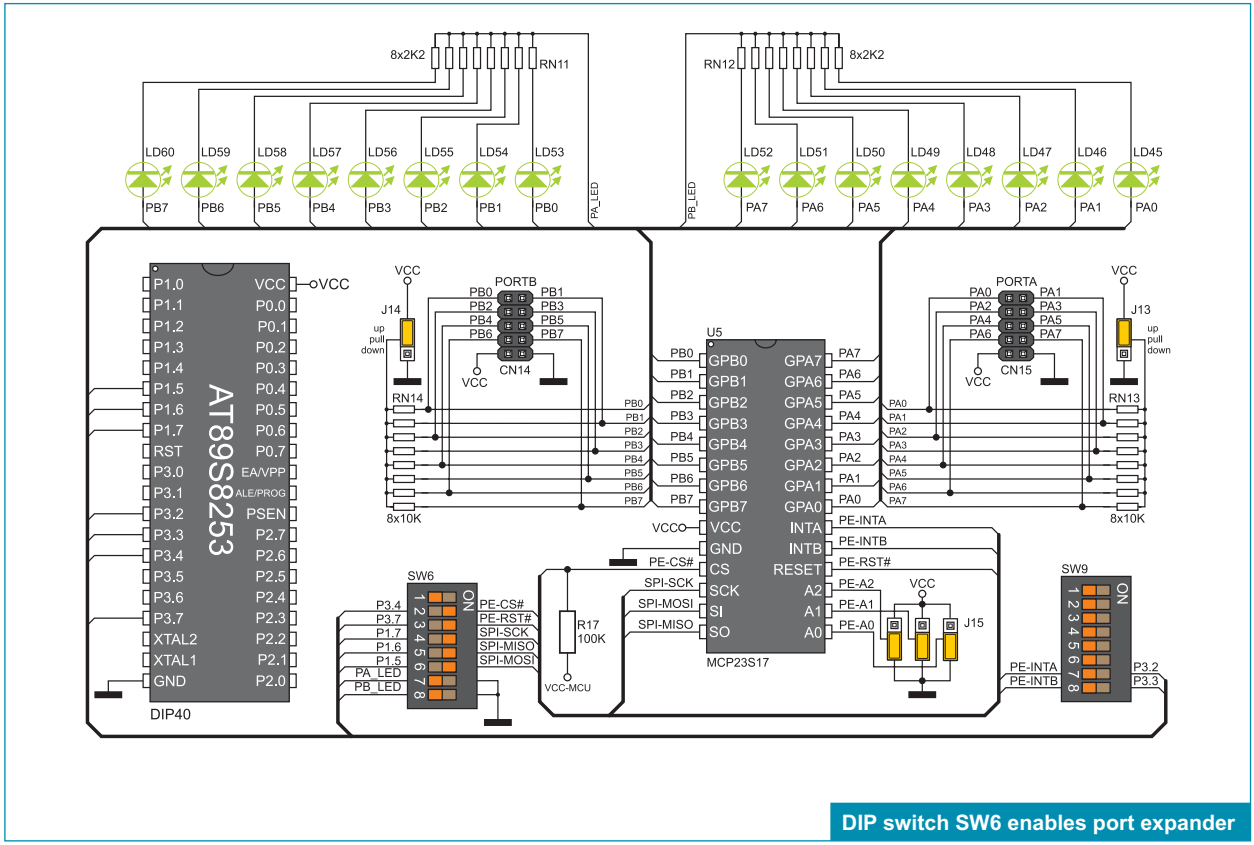


Figure 18-3: SPI communication block diagram

Jumpers J15 are used to determine the port expander's hardware address. These jumper are also used to provide the port expander's pins A2, A1 and A0 with a logic one (1) or zero (0). They should be placed in the low position (logic 0) by default.



DIP switch SW6 enables port expander

Figure 18-4: Port expander schematic

DISCLAIMER

All the products owned by MikroElektronika are protected by copyright law and international copyright treaty. Therefore, this manual is to be treated as any other copyright material. No part of this manual, including product and software described herein, may be reproduced, stored in a retrieval system, translated or transmitted in any form or by any means, without the prior written permission of MikroElektronika. The manual PDF edition can be printed for private or local use, but not for distribution. Any modification of this manual is prohibited.

MikroElektronika provides this manual 'as is' without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties or conditions of merchantability or fitness for a particular purpose.

MikroElektronika shall assume no responsibility or liability for any errors, omissions and inaccuracies that may appear in this manual. In no event shall MikroElektronika, its directors, officers, employees or distributors be liable for any indirect, specific, incidental or consequential damages (including damages for loss of business profits and business information, business interruption or any other pecuniary loss) arising out of the use of this manual or product, even if MikroElektronika has been advised of the possibility of such damages. MikroElektronika reserves the right to change information contained in this manual at any time without prior notice, if necessary.

HIGH RISK ACTIVITIES

The products of MikroElektronika are not fault – tolerant nor designed, manufactured or intended for use or resale as on – line control equipment in hazardous environments requiring fail – safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines or weapons systems in which the failure of Software could lead directly to death, personal injury or severe physical or environmental damage ('High Risk Activities'). MikroElektronika and its suppliers specifically disclaim any expressed or implied warranty of fitness for High Risk Activities.

TRADEMARKS

The Mikroelektronika name and logo, the Mikroelektronika logo, mikroC, mikroC PRO, mikroBasic, mikroBasic PRO, mikroPascal, mikroPascal PRO, AVRflash, PICflash, dsPICprog, 18FJprog, PSOCprog, AVRprog, 8051prog, ARMflash, EasyPIC5, EasyPIC6, BigPIC5, BigPIC6, dsPIC PRO4, Easy8051B, EasyARM, EasyAVR5, EasyAVR6, BigAVR2, EasydsPIC4A, EasyPSoC4, EasyVR Stamp LV18FJ, LV24-33A, LV32MX, PIC32MX4 MultiMedia Board, PICPLC16, PICPLC8 PICPLC4, SmartGSM/GPRS, UNI-DS are trademarks of Mikroelektronika. All other trademarks mentioned herein are property of their respective companies.

All other product and corporate names appearing in this manual may or may not be registered trademarks or copyrights of their respective companies, and are only used for identification or explanation and to the owners' benefit, with no intent to infringe.

©Mikroelektronika™, 2009, All Rights Reserved.

If you want to learn more about our products, please visit our website at www.mikroe.com

If you are experiencing some problems with any of our products or just need additional information, please place your ticket at www.mikroe.com/en/support

If you have any questions, comments or business proposals, do not hesitate to contact us at office@mikroe.com

TO OUR VALUED CUSTOMERS

I want to express my thanks to you for being interested in our products and for having confidence in mikroElektronika.
The primary aim of our company is to design and produce high quality electronic products and to constantly improve the performance thereof in order to better suit your needs.

Nebojsa Matic
General Manager

The Atmel name and logo, the Atmel logo, AVR, AVR (Logo), AVR Freaks, AVR Freaks (Logo), AVR Studio, IDIC, megaAVR, megaAVR (Logo), picoPower ®, tinyAVR ® are trademarks of Atmel Coorporation.

DISCLAIMER

All the products owned by MikroElektronika are protected by copyright law and international copyright treaty. Therefore, this manual is to be treated as any other copyright material. No part of this manual, including product and software described herein, may be reproduced, stored in a retrieval system, translated or transmitted in any form or by any means, without the prior written permission of MikroElektronika. The manual PDF edition can be printed for private or local use, but not for distribution. Any modification of this manual is prohibited.

MikroElektronika provides this manual 'as is' without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties or conditions of merchantability or fitness for a particular purpose.

MikroElektronika shall assume no responsibility or liability for any errors, omissions and inaccuracies that may appear in this manual. In no event shall MikroElektronika, its directors, officers, employees or distributors be liable for any indirect, specific, incidental or consequential damages (including damages for loss of business profits and business information, business interruption or any other pecuniary loss) arising out of the use of this manual or product, even if MikroElektronika has been advised of the possibility of such damages. MikroElektronika reserves the right to change information contained in this manual at any time without prior notice, if necessary.

HIGH RISK ACTIVITIES

The products of MikroElektronika are not fault – tolerant nor designed, manufactured or intended for use or resale as on – line control equipment in hazardous environments requiring fail – safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines or weapons systems in which the failure of Software could lead directly to death, personal injury or severe physical or environmental damage ('High Risk Activities'). MikroElektronika and its suppliers specifically disclaim any expressed or implied warranty of fitness for High Risk Activities.

TRADEMARKS

The Mikroelektronika name and logo, the Mikroelektronika logo, mikroC, mikroC PRO, mikroBasic, mikroBasic PRO, mikroPascal, mikroPascal PRO, AVRflash, PICflash, dsPICprog, 18FJprog, PSOCprog, AVRprog, 8051prog, ARMflash, EasyPIC5, EasyPIC6, BigPIC5, BigPIC6, dsPIC PRO4, Easy8051B, EasyARM, EasyAVR5, EasyAVR6, BigAVR2, EasydsPIC4A, EasyPSoC4, EasyVR Stamp LV18FJ, LV24-33A, LV32MX, PIC32MX4 MultiMedia Board, PICPLC16, PICPLC8 PICPLC4, SmartGSM/GPRS, UNI-DS are trademarks of Mikroelektronika. All other trademarks mentioned herein are property of their respective companies.

All other product and corporate names appearing in this manual may or may not be registered trademarks or copyrights of their respective companies, and are only used for identification or explanation and to the owners' benefit, with no intent to infringe.

©Mikroelektronika™, 2009, All Rights Reserved.

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

MikroElektronika:

MIKROE-455